WealthData Analysis and Forecast

Sally Chen

Contents


```
library(fpp)
library(fpp2)
library(forecast)
```

```
temp = read.csv("MonthTemp.csv")
```

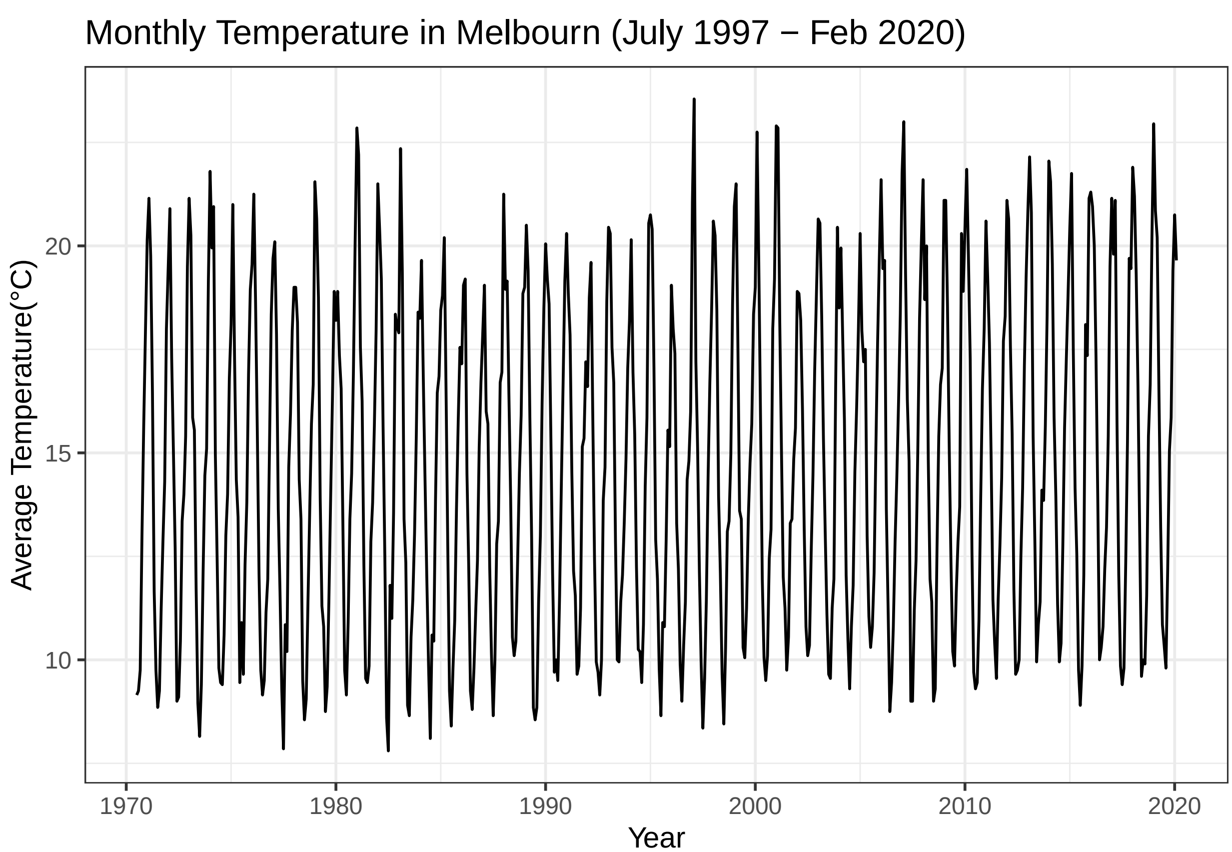
```
temp$Mean = (temp$Min + temp$Max)/2

tempts_mean = ts(temp[,5], frequency = 12, start = c(1970,7), names = "temp_mean") #
↪  transform to time series data

summary(temp)
```

```
##       Year          Month              Min               Max
##  Min.   :1970   Min.   : 1.000   Min.   : 3.100   Min.   :11.70
##  1st Qu.:1982   1st Qu.: 3.750   1st Qu.: 6.800   1st Qu.:15.38
##  Median :1995   Median : 7.000   Median : 9.200   Median :19.95
##  Mean   :1995   Mean   : 6.513   Mean   : 9.583   Mean   :19.92
##  3rd Qu.:2007   3rd Qu.:10.000   3rd Qu.:12.200   3rd Qu.:24.40
##  Max.   :2020   Max.   :12.000   Max.   :16.800   Max.   :30.40
##       Mean
##  Min.   : 7.80
##  1st Qu.:11.15
##  Median :14.50
##  Mean   :14.75
##  3rd Qu.:18.30
##  Max.   :23.55
```

Graphical exploration of the data.

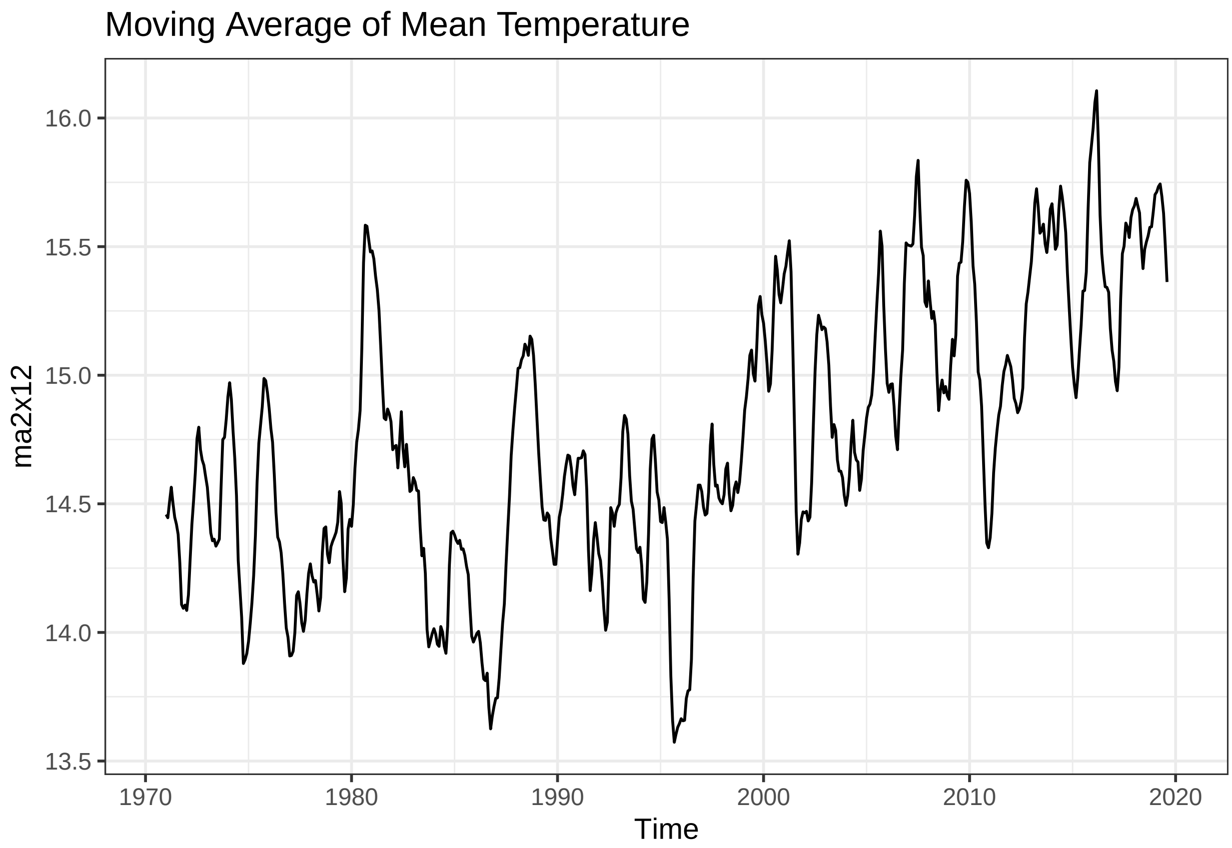
Time series plot

```
autoplot(tempts_mean, main = "Monthly Temperature in Melbourn (July 1997 - Feb 2020)",
↪  ylab = "Average Temperature(\u00B0C)", xlab = "Year") + theme_bw()
```



Moving average (ma)

```
ma2x12 <- ma(tempts_mean, order=12, centre=TRUE)
autoplot(ma2x12, main = "Moving Average of Mean Temperature") + theme_bw()
```

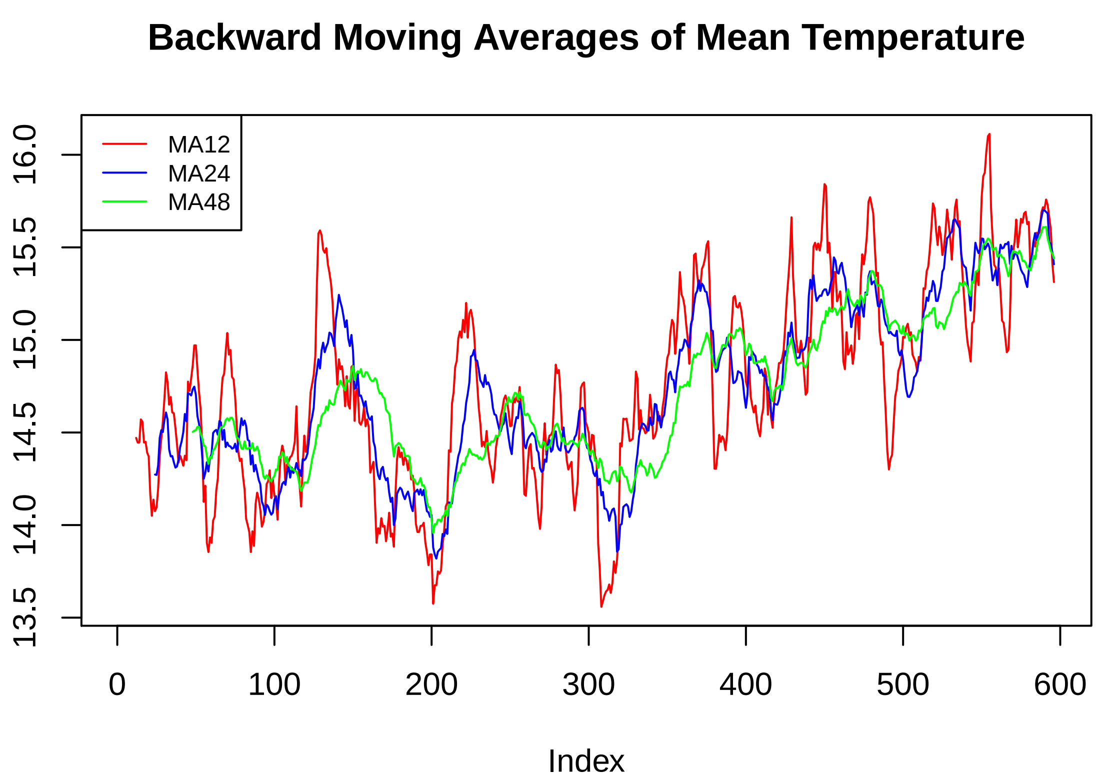


End-point ma to show trend

```
#define a function for backwards MA
ma_bk=function(y, order, ...){
  ma<-matrix(0,nrow(y),1)
  for(i in order:nrow(y))
  {
    ma[i]<-mean(y[i-order+1:order])
  }
  ma[1:(order-1)]=NA
  return(ma)
}
```

```
tp_mean =as.matrix(temp$Mean)

tp_ma12=ma_bk(tp_mean,12) # number of week days
```

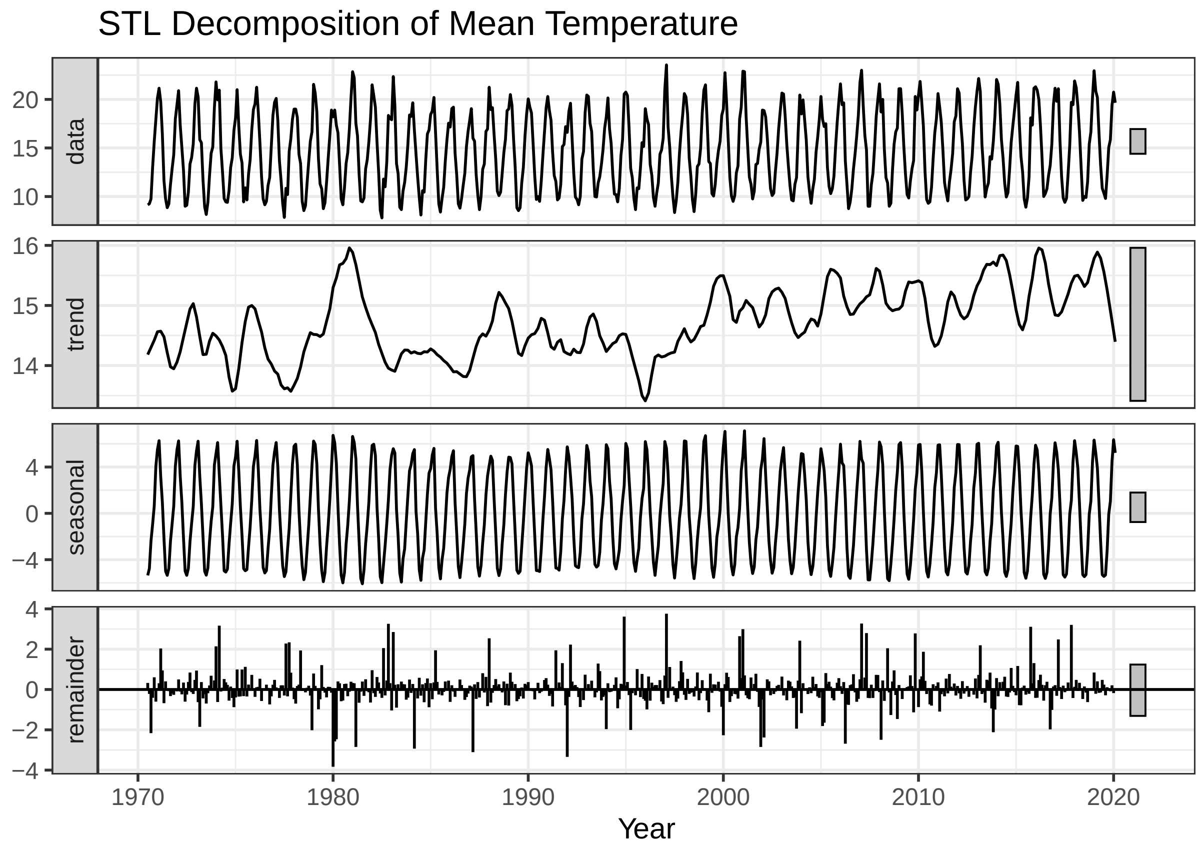
```
plot(tp_ma12,type="l", col="red", main = "Backward Moving Averages of Mean Temperature",
↪  ylab = "")

tp_ma24=ma_bk(tp_mean,24)
lines(tp_ma24,col="blue")

tp_ma48=ma_bk(tp_mean,48)
lines(tp_ma48,col="green")

legend("topleft", legend = c("MA12", "MA24", "MA48"), col = c("red", "blue", "green"),
↪  lty = 1, cex = 0.75)
```



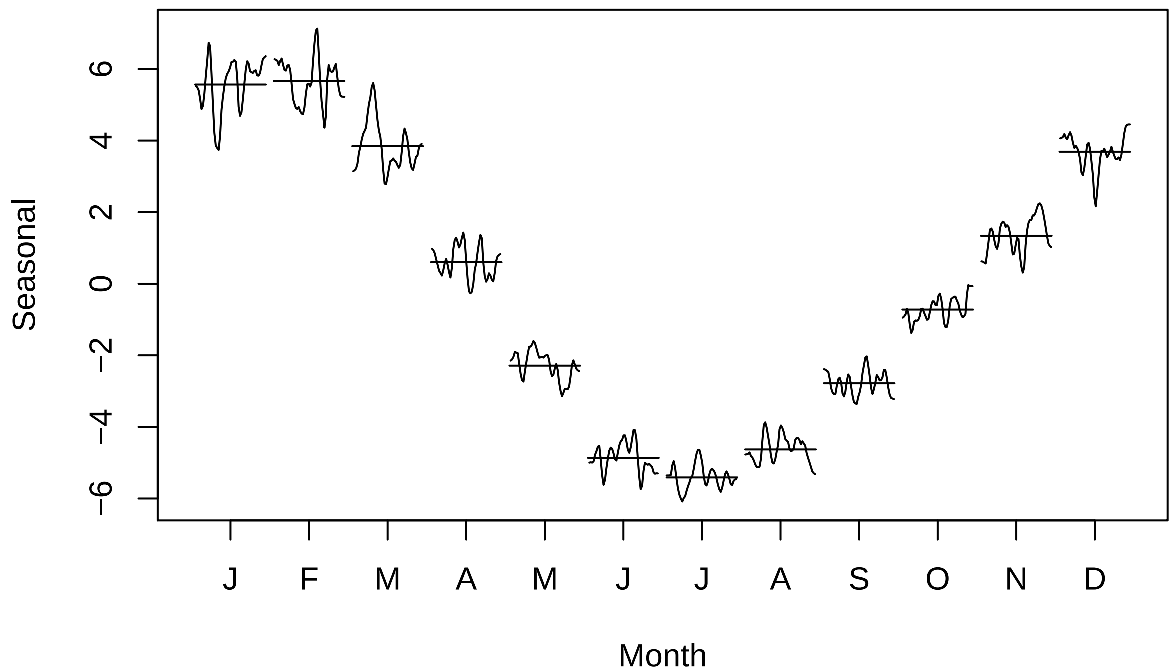
Fluctuation, intersection, increasing and decreasing trend.

Time series decomposition of the data - stl

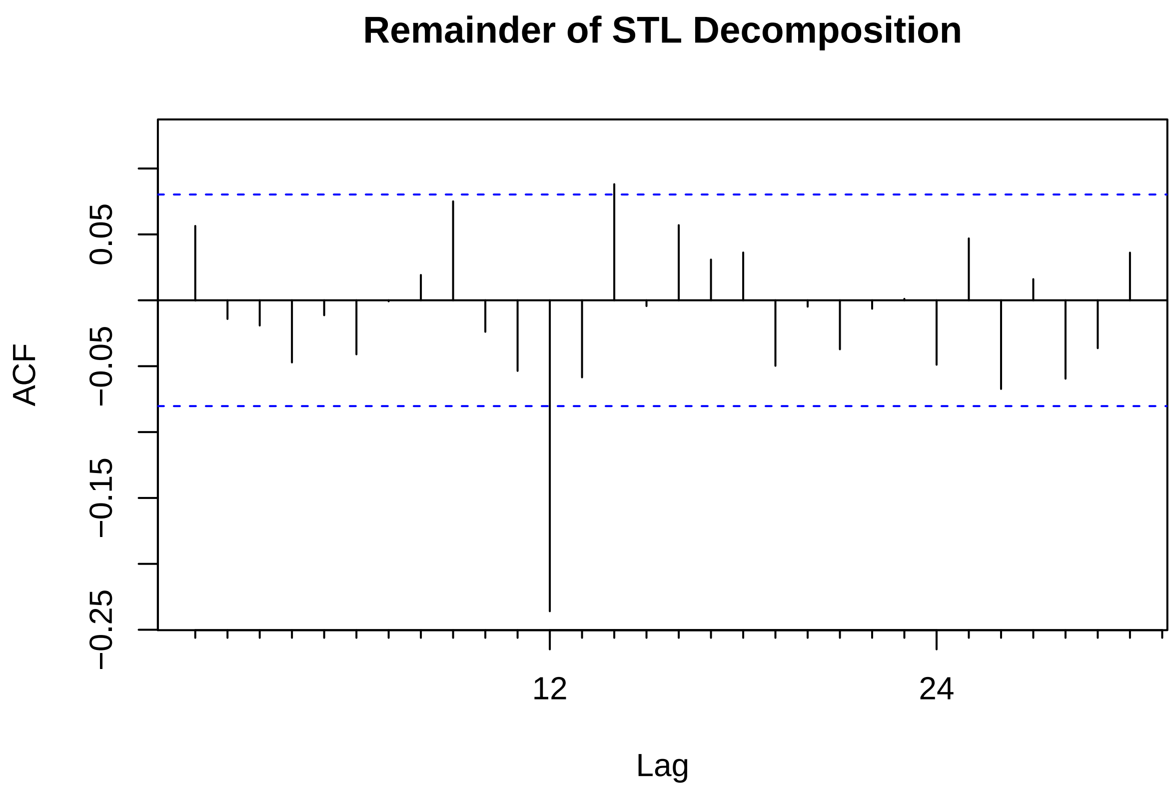
```
fitstl = tempts_mean %>% stl(t.window = 12, s.window = 6, robust = TRUE) # STL
↪  decomposition
fitstl%>% autoplot()+ xlab("Year") +
  ggtitle("STL Decomposition of Mean Temperature") +theme_bw()
```




```
monthplot(fitstl$time.series[, "seasonal"], main = "", ylab = "Seasonal", xlab = "Month")
```



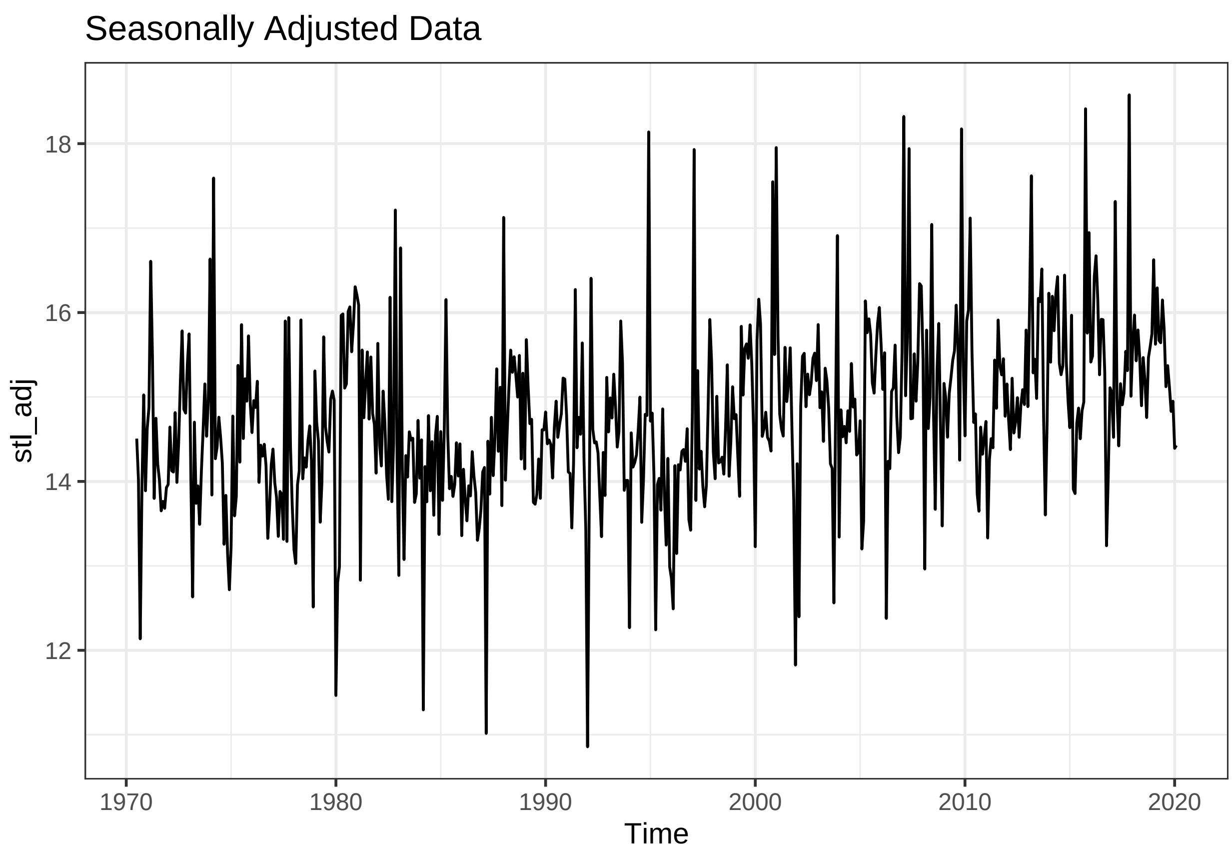

```
stl_res = remainder(fitstl)
Acf(stl_res,lag.max=30, main = "Remainder of STL Decomposition")
```




```
Box.test(stl_res,type="Lj")
# dwtest(fitstl,alt="two.sided")
# bgtest(fitstl,10)
```

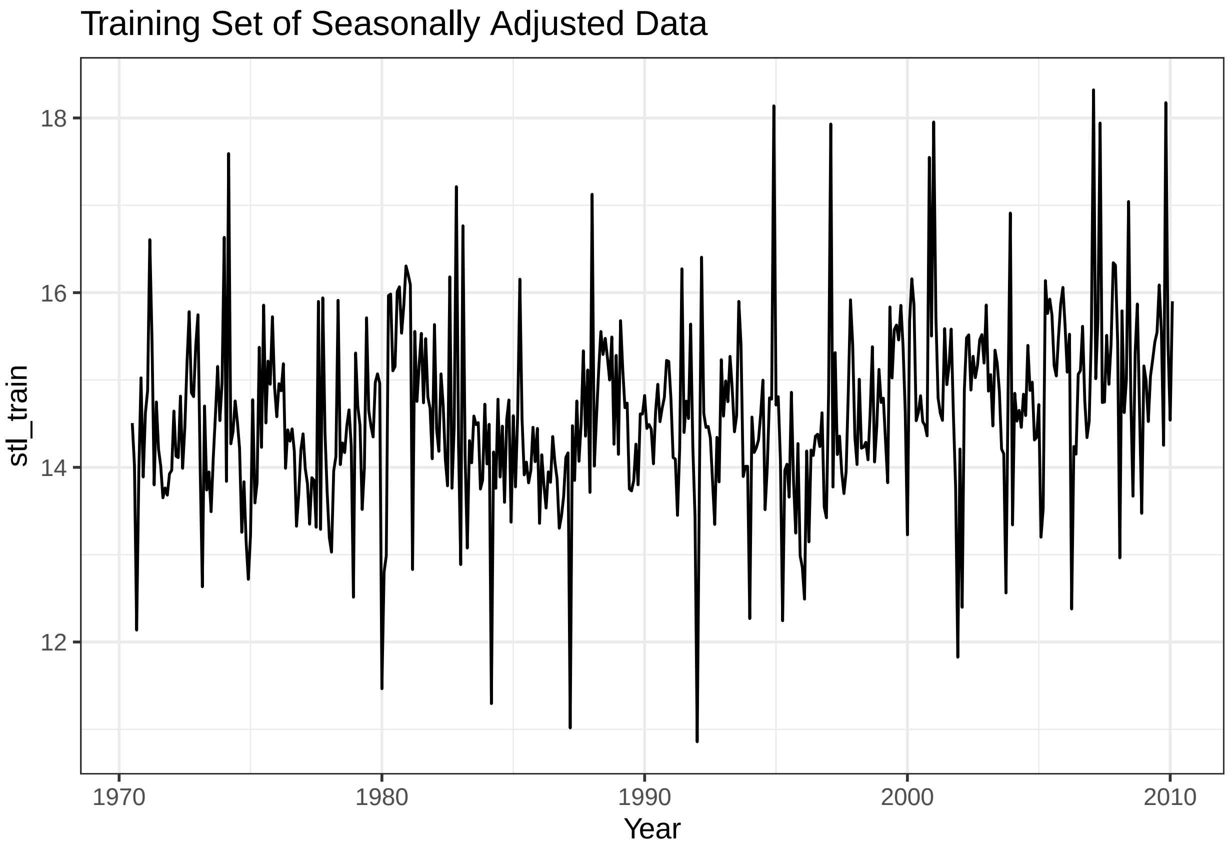
```
##
##  Box-Ljung test
##
## data:  stl_res
## X-squared = 1.9062, df = 1, p-value = 0.1674
```

```
stl_adj = seasadj(fitstl)
autoplot(stl_adj, main = "Seasonally Adjusted Data") +theme_bw()
```

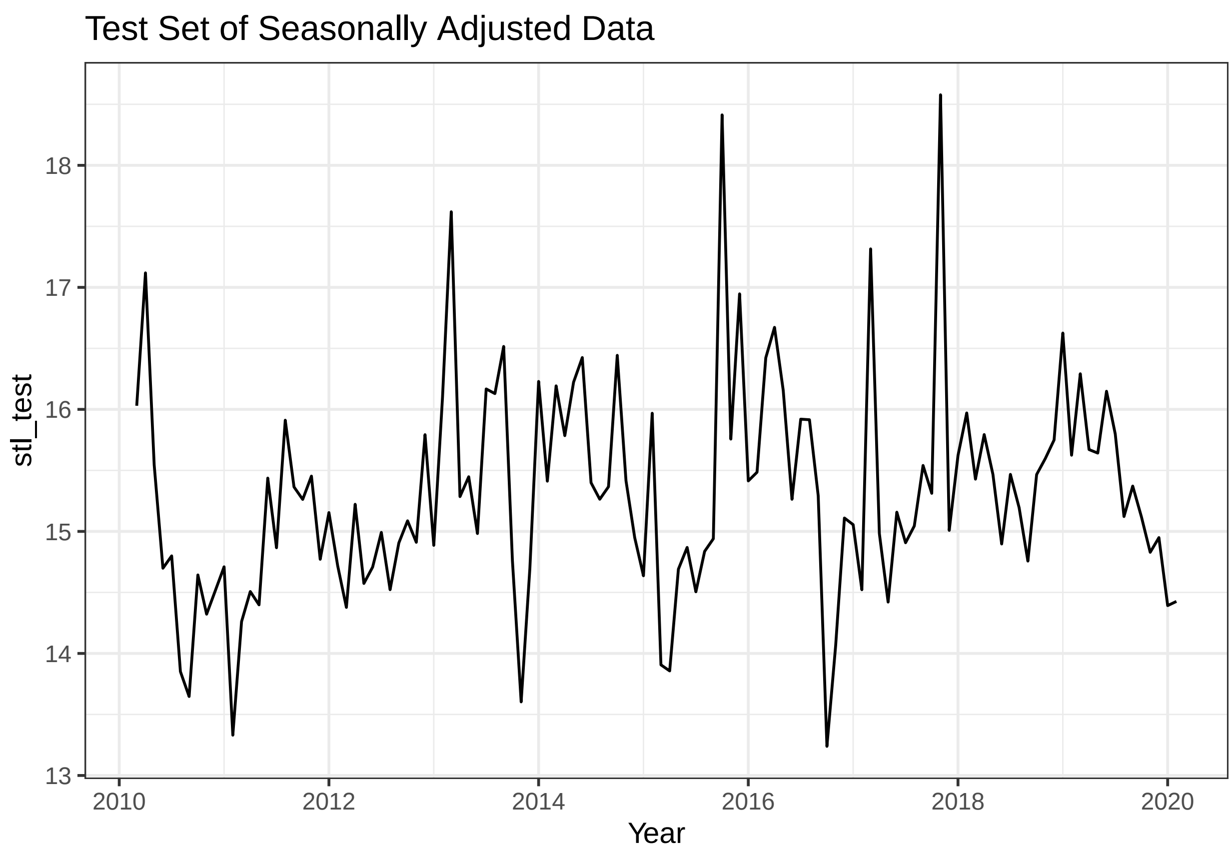


Split into training and test set

```
# Split data using 80-20 rule
stl_train = window(stl_adj, end = c(2010,2))
autoplot(stl_train, main = "Training Set of Seasonally Adjusted Data", xlab = "Year")
↪   +theme_bw()
```




```
stl_test = window(stl_adj, start = c(2010,3))
autoplot(stl_test, main = "Test Set of Seasonally Adjusted Data", xlab = "Year")
↪   +theme_bw()
```



Regression analysis

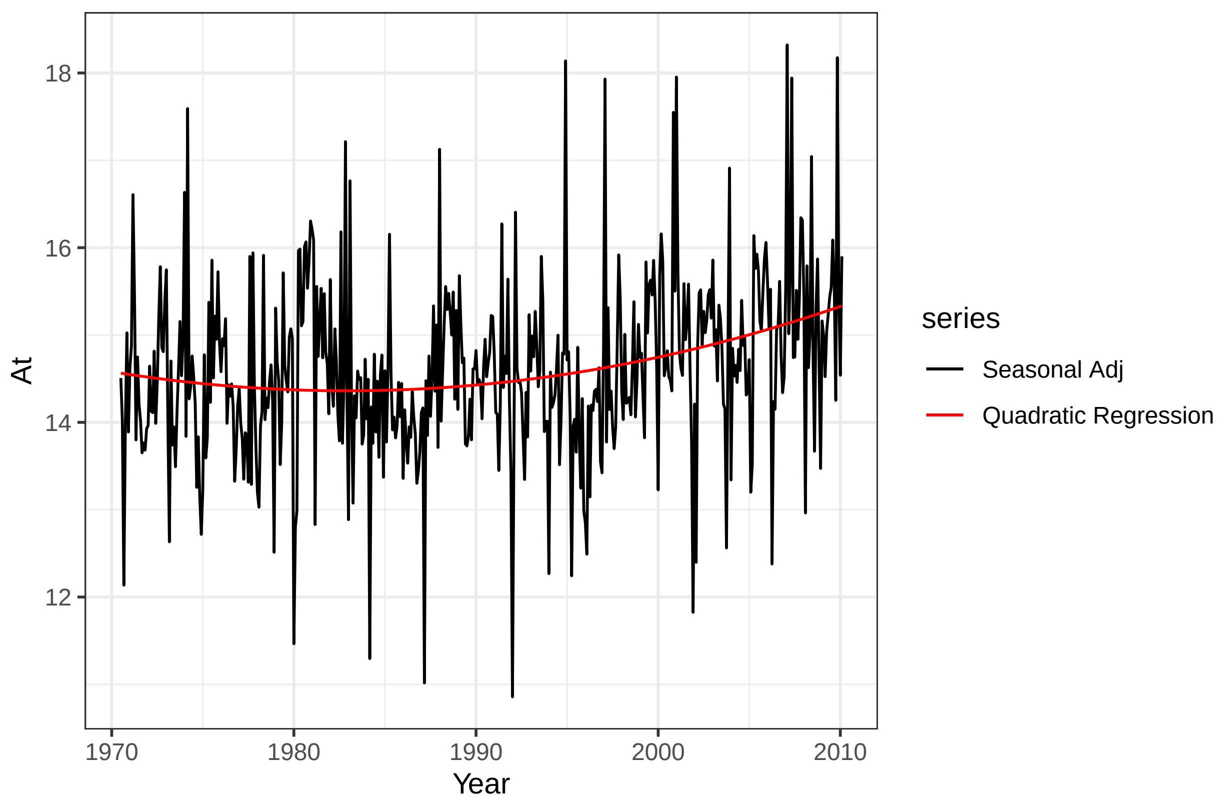
Quadratic regression

```
fit_q = tslm(stl_train ~ trend + I(trend^2))
summary(fit_q)
```

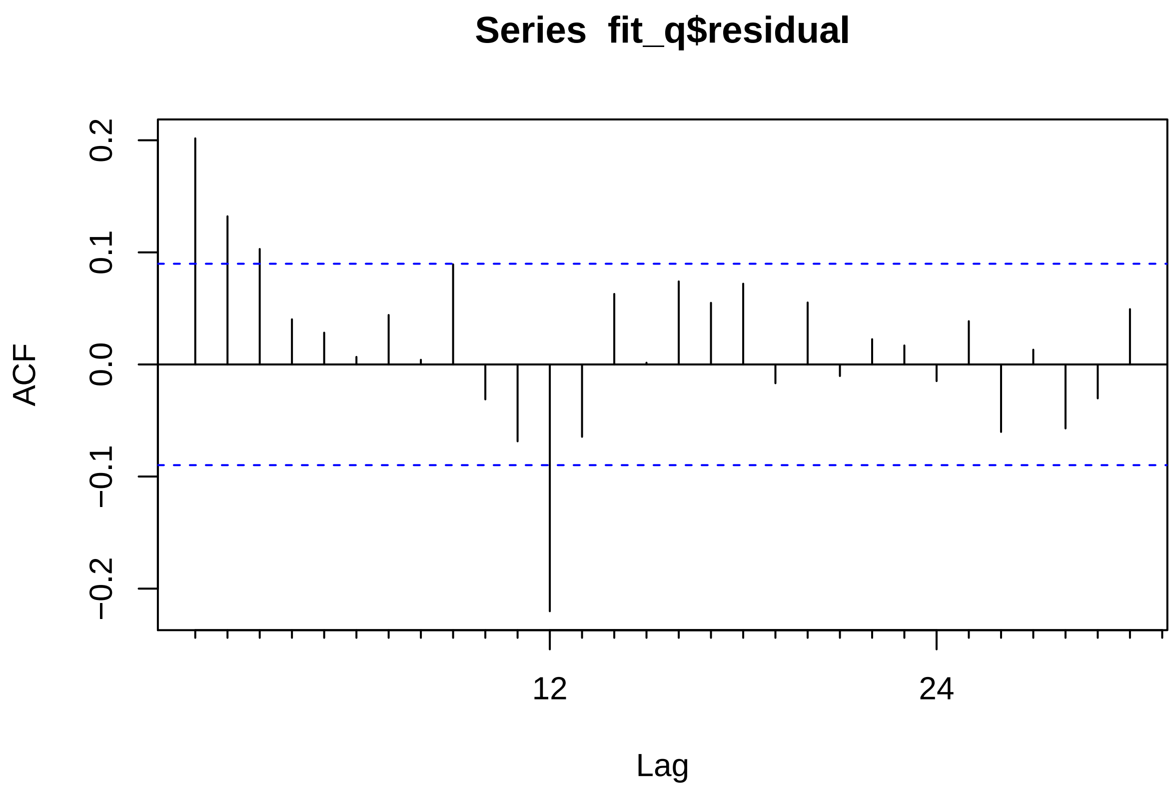
```
##
## Call:
## tslm(formula = stl_train ~ trend + I(trend^2))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6122 -0.5310 -0.0245  0.4566  3.5882
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.457e+01  1.344e-01 108.398  < 2e-16 ***
## trend       -2.739e-03  1.301e-03  -2.105 0.035799 *
## I(trend^2)   9.136e-06  2.641e-06   3.459 0.000591 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.9732 on 473 degrees of freedom
## Multiple R-squared:  0.07227,    Adjusted R-squared:  0.06835
## F-statistic: 18.42 on 2 and 473 DF,  p-value: 1.973e-08
```

```
autoplot(stl_train, series = "Seasonal Adj", ylab = "At", xlab = "Year") +
↪   autolayer(fitted(fit_q), series = "Quadratic Regression") + scale_color_manual(values
↪   = c("black","red"),                                                 breaks =
↪   c("Seasonal Adj","Quadratic Regression")) + theme_bw()
```




```
CV(fit_q)
Acf(fit_q$residual,lag.max=30)
```




```r
Box.test(fit_q$residual, fitdf=length(fit_q$coefficients)+1,lag=10,type="Lj")
dwtest(fit_q,alt="two.sided")
bgtest(fit_q,10)
```

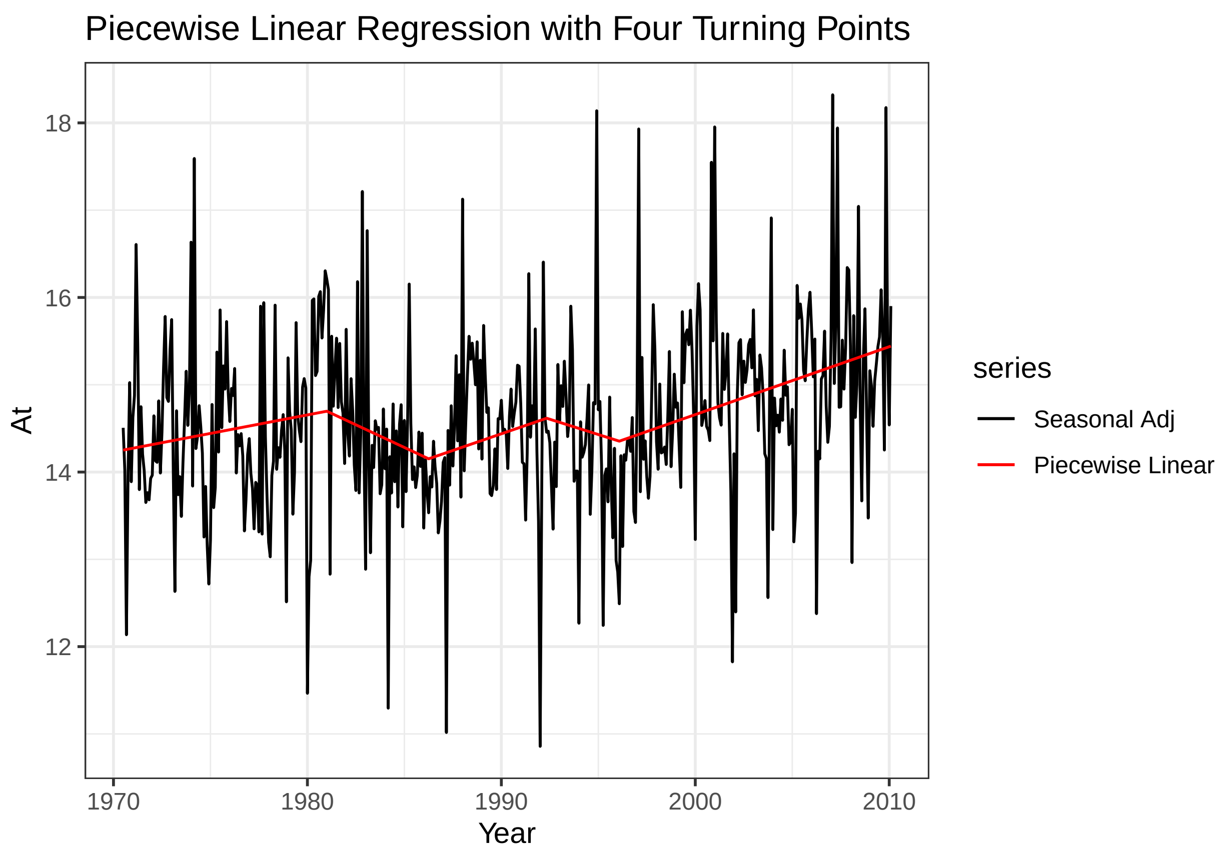
```
##           CV          AIC         AICc          BIC        AdjR2
##   0.95306024 -20.91900163 -20.83407594  -4.25733021   0.06834733
##
##  Box-Ljung test
##
## data:  fit_q$residual
## X-squared = 39.486, df = 6, p-value = 5.747e-07
##
##
##  Durbin-Watson test
##
## data:  fit_q
## DW = 1.5958, p-value = 6.462e-06
## alternative hypothesis: true autocorrelation is not 0
##
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  fit_q
## LM test = 32.167, df = 10, p-value = 0.0003756
```

Piecewise linear 4 turning points

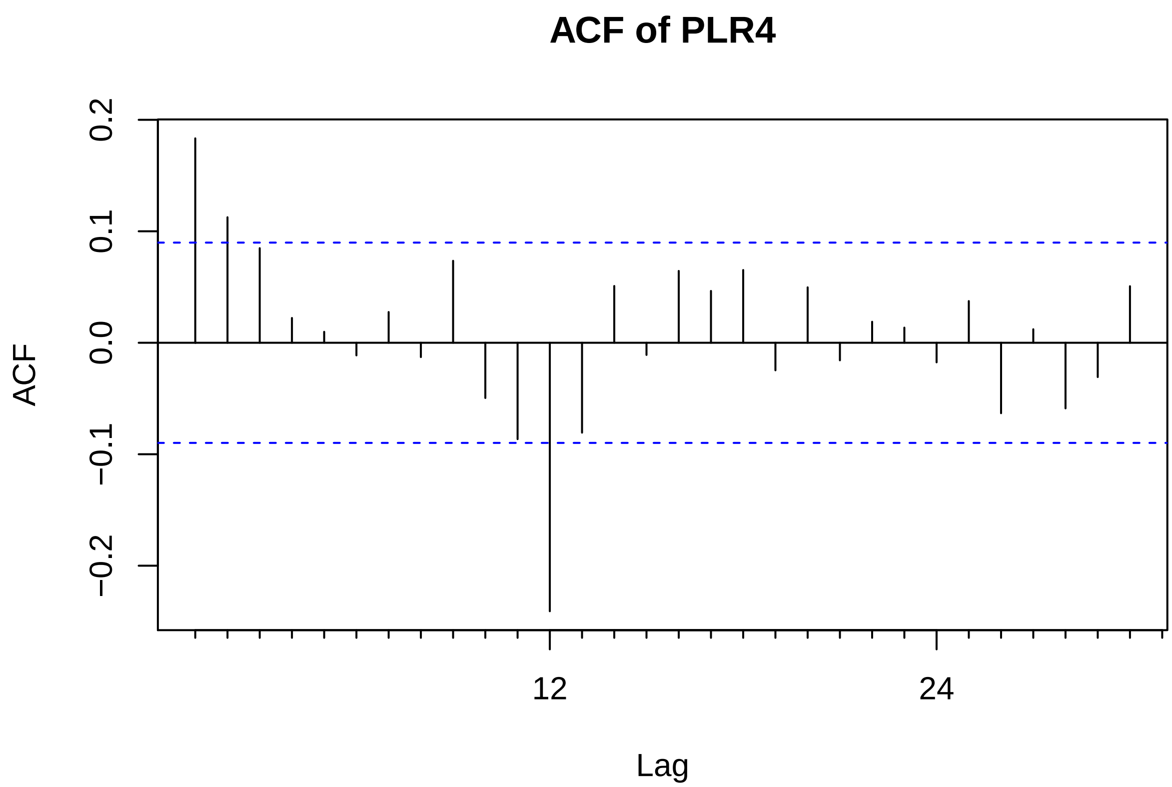
```r
fit_pl4 = tslm(stl_train~trend+I(pmax(trend-127,0)) + I(pmax(trend-190,0)) +
↪  I(pmax(trend-263,0)) +  I(pmax(trend-308,0)))

summary(fit_pl4)
CV(fit_pl4)
```

```
##
## Call:
## tslm(formula = stl_train ~ trend + I(pmax(trend - 127, 0)) +
##     I(pmax(trend - 190, 0)) + I(pmax(trend - 263, 0)) + I(pmax(trend -
##     308, 0)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7336 -0.4800 -0.0584  0.4615  3.7034
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             14.248024   0.164427  86.652   <2e-16 ***
## trend                    0.003534   0.002000   1.767   0.0779 .
## I(pmax(trend - 127, 0)) -0.012187   0.005082  -2.398   0.0169 *
## I(pmax(trend - 190, 0))  0.015020   0.006445   2.331   0.0202 *
## I(pmax(trend - 263, 0)) -0.012207   0.007790  -1.567   0.1178
## I(pmax(trend - 308, 0))  0.012318   0.005902   2.087   0.0374 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9651 on 470 degrees of freedom
## Multiple R-squared:  0.0933, Adjusted R-squared:  0.08366
## F-statistic: 9.673 on 5 and 470 DF,  p-value: 8.348e-09
##
##           CV          AIC         AICc          BIC        AdjR2
##   0.94355662 -25.83388344 -25.59456720   3.32404154   0.08365554
```

```r
autoplot(stl_train, main = 'Piecewise Linear Regression with Four Turning Points', ylab =
↪  'At', series = "Seasonal Adj", xlab = "Year") + autolayer(fitted(fit_pl4), series =
↪  'Piecewise Linear') +
  scale_color_manual(values = c("black","red"),
                     breaks = c("Seasonal Adj","Piecewise Linear")) + theme_bw()
```




```
Acf(fit_pl4$residual,lag.max=30, main = "ACF of PLR4")
```




```r
Box.test(fit_pl4$residual, fitdf=length(fit_pl4$coefficients)+1,lag=10,type="Lj")
dwtest(fit_pl4,alt="two.sided")
bgtest(fit_pl4,10)
```

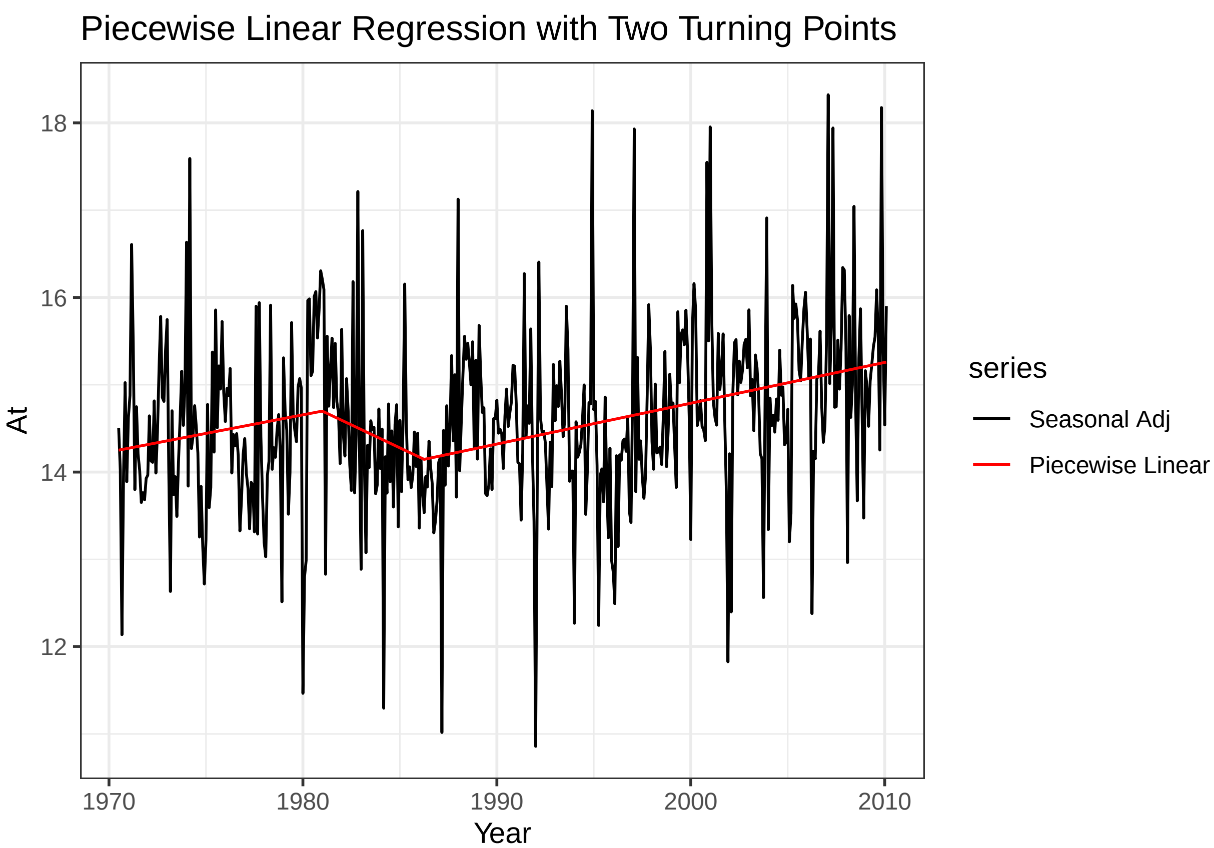
```
##
##  Box-Ljung test
##
## data:  fit_pl4$residual
## X-squared = 30.287, df = 3, p-value = 1.201e-06
##
##
##  Durbin-Watson test
##
## data:  fit_pl4
## DW = 1.6326, p-value = 2.143e-05
## alternative hypothesis: true autocorrelation is not 0
##
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  fit_pl4
## LM test = 27.211, df = 10, p-value = 0.002411
```

Piecewise linear 2 turning points

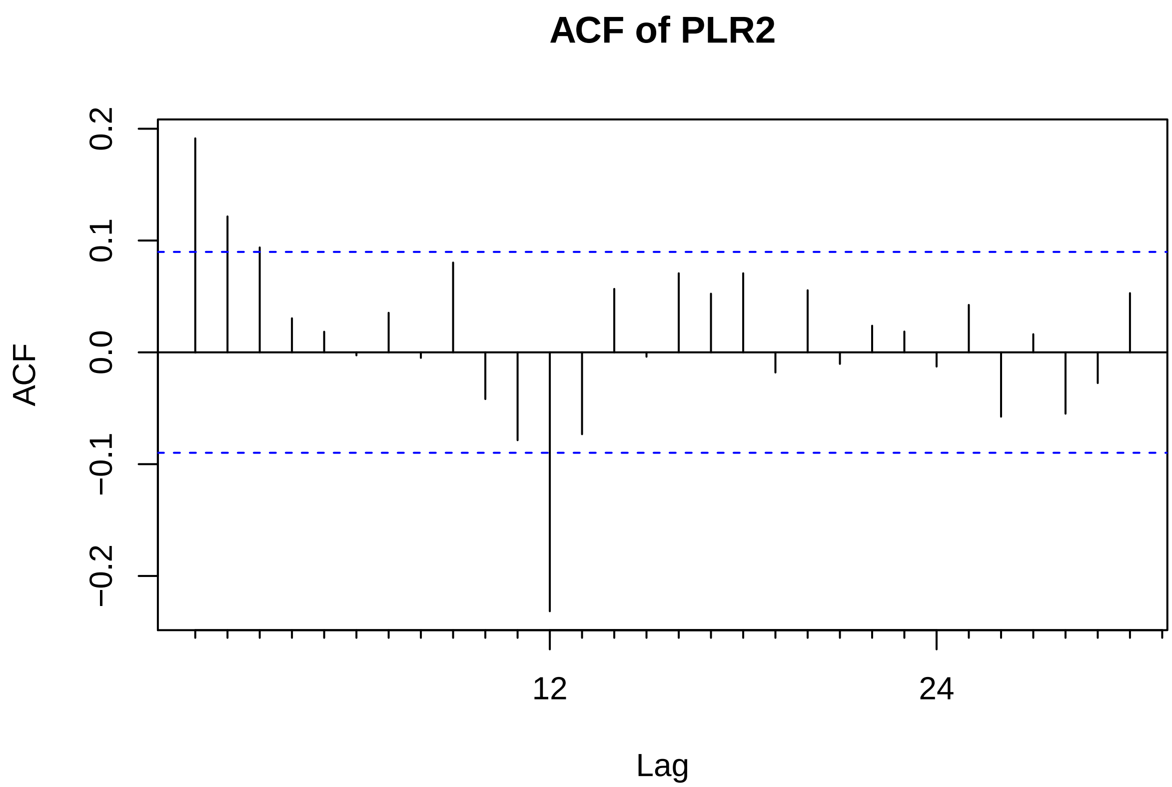
```
fit_pl2 = tslm(stl_train~trend+I(pmax(trend-127,0)) + I(pmax(trend-190,0)))
# + I(pmax(trend-263,0)) + I(pmax(trend-308,0))
summary(fit_pl2)
CV(fit_pl2)
```

```
##
## Call:
## tslm(formula = stl_train ~ trend + I(pmax(trend - 127, 0)) +
##     I(pmax(trend - 190, 0)))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.5568 -0.4898 -0.0374  0.4675  3.5882
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)             14.247391   0.164504  86.608  < 2e-16 ***
## trend                    0.003549   0.001987   1.786 0.074795 .
## I(pmax(trend - 127, 0)) -0.012321   0.004452  -2.767 0.005871 **
## I(pmax(trend - 190, 0))  0.012668   0.003272   3.872 0.000123 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.968 on 472 degrees of freedom
## Multiple R-squared:  0.08399,    Adjusted R-squared:  0.07816
## F-statistic: 14.43 on 3 and 472 DF,  p-value: 5.252e-09
##
##           CV          AIC         AICc          BIC        AdjR2
##   0.94522594 -24.96822504 -24.84056546  -4.14113576   0.07816336
```

```
autoplot(stl_train, main = 'Piecewise Linear Regression with Two Turning Points', ylab =
↪  'At', series = "Seasonal Adj", xlab = "Year") + autolayer(fitted(fit_pl2), series =
↪  'Piecewise Linear') +
  scale_color_manual(values = c("black","red"),
                     breaks = c("Seasonal Adj","Piecewise Linear")) + theme_bw()
```




```
Acf(fit_pl2$residual,lag.max=30, main = "ACF of PLR2")
```




```r
Box.test(fit_pl2$residual, fitdf=length(fit_pl2$coefficients)+1,lag=10,type="Lj")
dwtest(fit_pl2,alt="two.sided")
bgtest(fit_pl2,10)
```

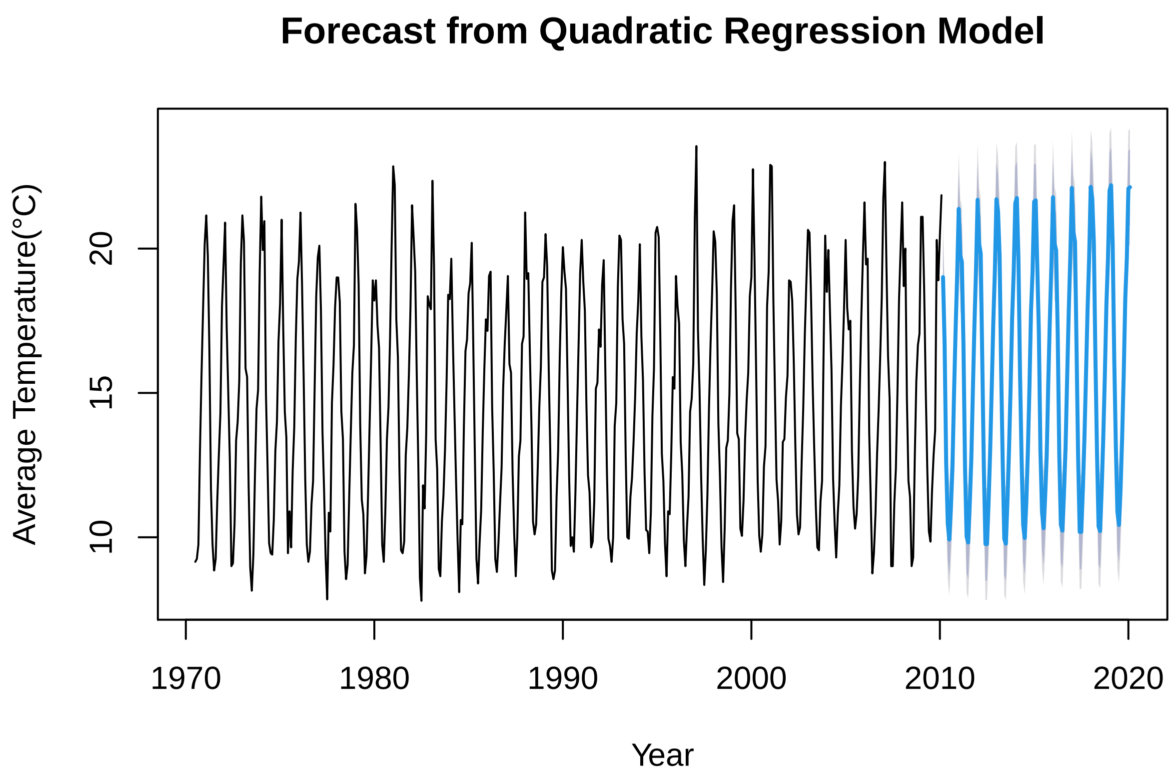
```
##
##  Box-Ljung test
##
## data:  fit_pl2$residual
## X-squared = 34.103, df = 5, p-value = 2.271e-06
##
##
##  Durbin-Watson test
##
## data:  fit_pl2
## DW = 1.6161, p-value = 1.454e-05
## alternative hypothesis: true autocorrelation is not 0
##
##
##  Breusch-Godfrey test for serial correlation of order up to 10
##
## data:  fit_pl2
## LM test = 29.349, df = 10, p-value = 0.001094
```

Forecast evaluation

```
tp_in = window(tempts_mean, end = c(2010,2))
tp_out = window(tempts_mean, start = c(2010,3))
season_in = (tp_in - stl_train)[-c(1:416)] #extract 5yr seasonal pattern from latest
↪  train dataset
season_rep = rep(season_in,2) # replicate seasonal pattern to 10yr
```

Quadratic regression

```
fcast_q = forecast(fit_q, h = 120)

fcast_q$x = fcast_q$x + (tp_in - stl_train) # original
fcast_q$mean = fcast_q$mean + season_rep # forecast
fcast_q$upper = fcast_q$upper + season_rep
fcast_q$lower = fcast_q$lower + season_rep

plot(fcast_q, main = "Forecast from Quadratic Regression Model", ylab = "Average
↪  Temperature(\u00B0C)", xlab = "Year")
```

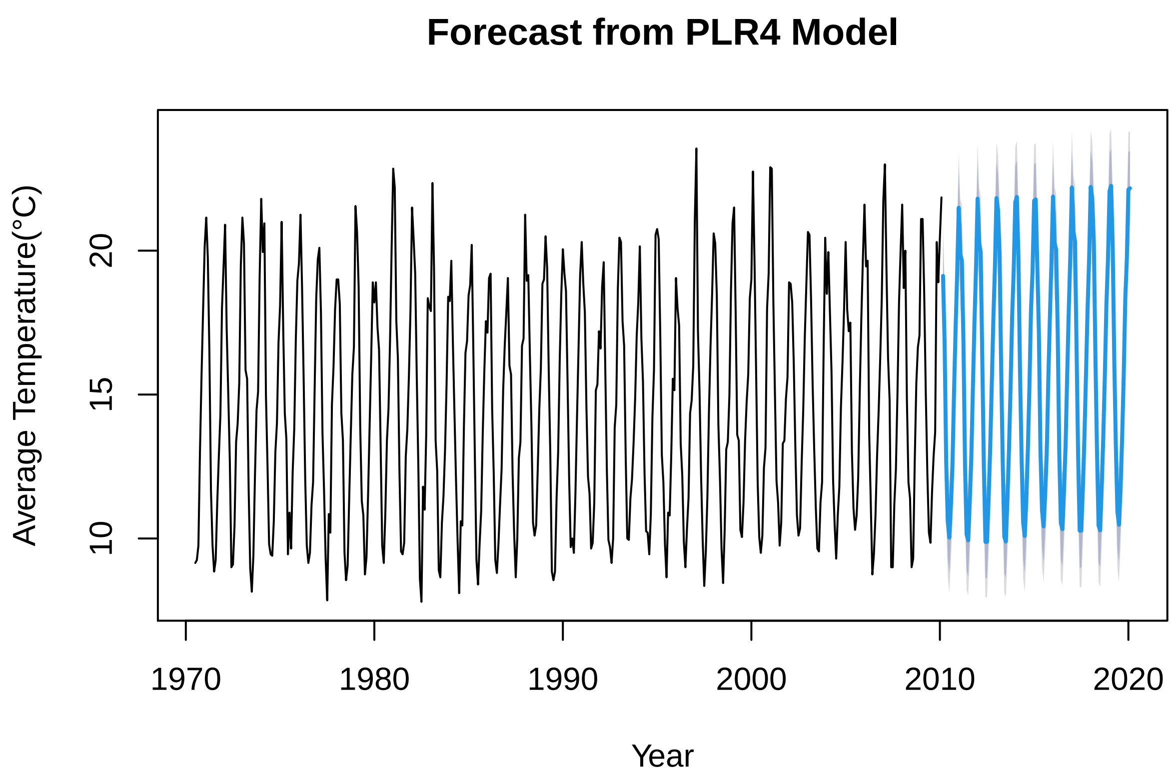


```r
accuracy(fcast_q, stl_test+season_rep)
```

```
##                        ME      RMSE       MAE       MPE      MAPE     MASE
## Training set  0.007357466 3.9526598 3.4417170 -7.965914 25.903958 2.991116
## Test set     -0.427073768 0.9748079 0.7952625 -3.415669  5.518487 0.691144
##                   ACF1 Theil's U
## Training set 0.8129081        NA
## Test set     0.3105289 0.4236062
```

Piecewise linear 4 turning points

```r
fcast_pl4 = forecast(fit_pl4, h = 120)

fcast_pl4$x = fcast_pl4$x + (tp_in - stl_train) # original
fcast_pl4$mean = fcast_pl4$mean + season_rep # forecast
fcast_pl4$upper = fcast_pl4$upper + season_rep
fcast_pl4$lower = fcast_pl4$lower + season_rep

plot(fcast_pl4, main = "Forecast from PLR4 Model", ylab = "Average Temperature(\u00B0C)",
↪  xlab = "Year")
```

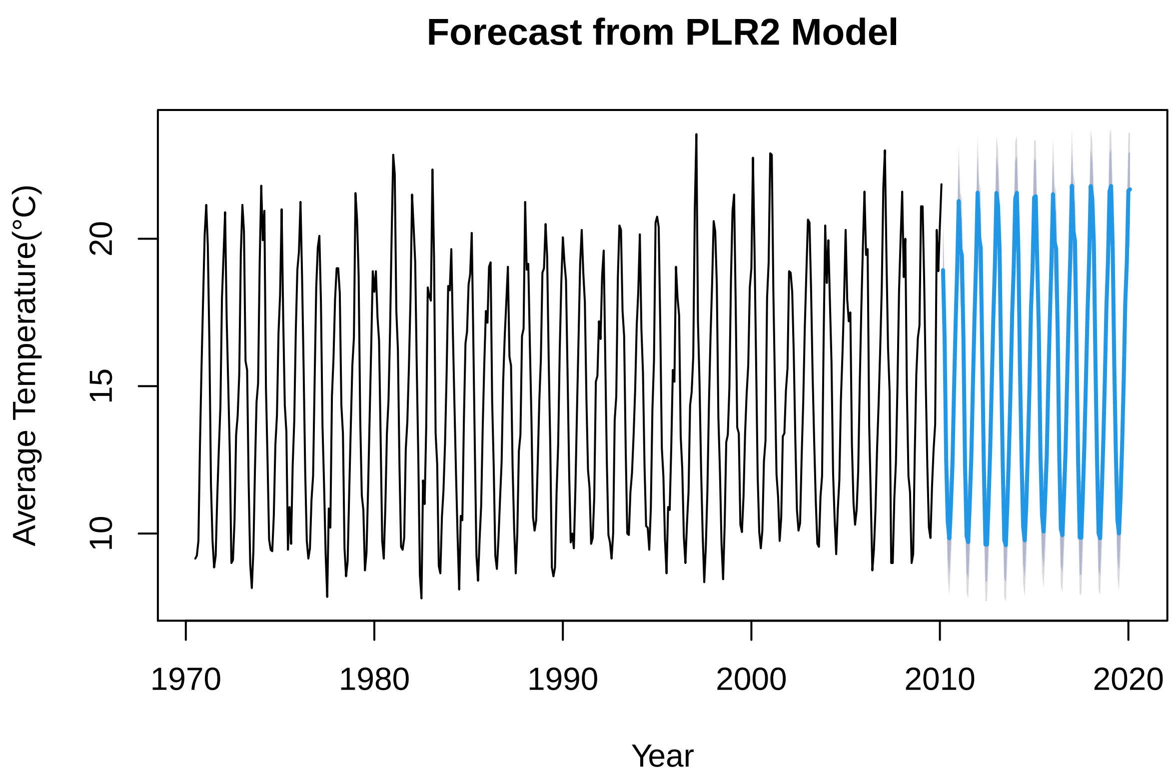


```r
accuracy(fcast_pl4, stl_test+season_rep)
```

```
##                        ME     RMSE       MAE       MPE     MAPE      MASE
## Training set  0.007357466 3.948894 3.4397866 -7.953892 25.88923 2.9894381
## Test set     -0.523642835 1.018083 0.8437648 -4.101302  5.90031 0.7332962
##                   ACF1 Theil's U
## Training set 0.8128762        NA
## Test set     0.3064732 0.4461339
```

Piecewise linear 2 turning points

```r
fcast_pl2 = forecast(fit_pl2, h = 120)

fcast_pl2$x = fcast_pl2$x + (tp_in - stl_train) # original
fcast_pl2$mean = fcast_pl2$mean + season_rep # forecast
fcast_pl2$upper = fcast_pl2$upper + season_rep
fcast_pl2$lower = fcast_pl2$lower + season_rep

plot(fcast_pl2, main = "Forecast from PLR2 Model", ylab = "Average Temperature(\u00B0C)",
↪  xlab = "Year")
```



```r
accuracy(fcast_pl2, stl_test+season_rep)
```

```
##                        ME      RMSE       MAE       MPE      MAPE      MASE
## Training set  0.007357466 3.9506512 3.4431170 -7.959386 25.911792 2.9923324
## Test set     -0.184961692 0.8915683 0.6825558 -1.722182  4.642317 0.5931932
##                   ACF1 Theil's U
## Training set 0.8128910        NA
## Test set     0.3088086 0.3792033
```

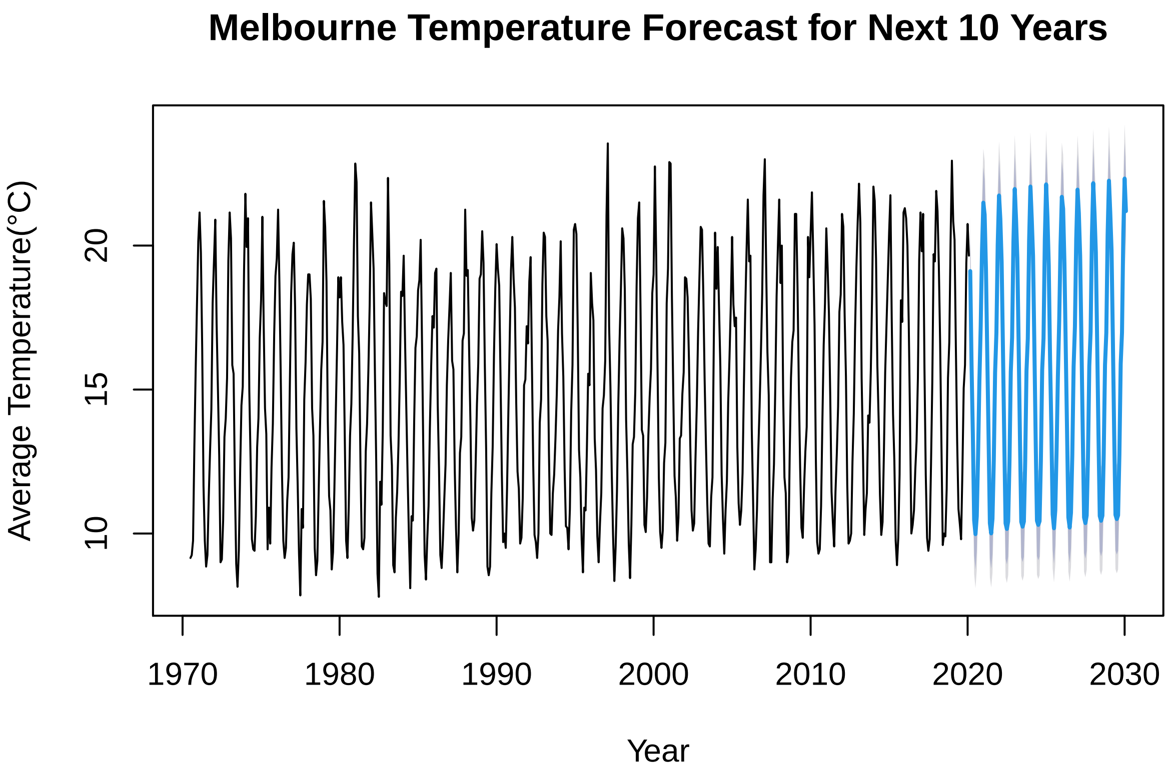
Compare Forecast Error Difference

```r
# Compare Forecast Error Difference between 2 piecewise linear models
dm.test((tp_out-fcast_pl2$mean),(tp_out-fcast_pl4$mean),power=2, alternative = "l")
```

```
##
##  Diebold-Mariano Test
##
## data:  (tp_out - fcast_pl2$mean)(tp_out - fcast_pl4$mean)
## DM = -4.0166, Forecast horizon = 1, Loss function power = 2, p-value =
## 5.19e-05
## alternative hypothesis: less
```

Forcast Temperature for next 10 years

```r
# Use piecewise linear 2 turning points, since it has best forecast accuracy
n10_fit = tslm(stl_adj~trend+I(pmax(trend-127,0)) + I(pmax(trend-190,0))) # fit model
↪  with whole dataset
n10_fcast = forecast(n10_fit, h = 120)

seasonC = (tempts_mean - stl_adj)[-c(1:536)]
seasonC_rep = rep(seasonC,2)

n10_fcast$x = n10_fcast$x + (tempts_mean - stl_adj) # original
n10_fcast$mean = n10_fcast$mean + seasonC_rep # forecast
n10_fcast$upper = n10_fcast$upper + seasonC_rep
n10_fcast$lower = n10_fcast$lower + seasonC_rep

plot(n10_fcast, main = "Melbourne Temperature Forecast for Next 10 Years", ylab =
↪  "Average Temperature(\u00B0C)", xlab = "Year")
```


Melbourne Temperature Forecast for Next 10 Years
Average Temperature(°C)
10
15
20
1970
1980
1990
2000
2010
2020
2030
Year